OIPE
NOV 16 2006
PATENT & TRADEMARK OFFICE

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1.-3. (canceled)

4. (currently amended) A method according to claim ~~3~~23, wherein a new address token is issued to said client application if said authentication data is invalid.

5. (currently amended) A method according to claim 4, wherein said address token further comprises data indicating the number of times an invalid authenticator has been received from said client application.

6. (previously presented) A method according to claim 5, wherein said method comprises transmitting no further address token to said client application if an address token received from said client application indicates that a predetermined number of invalid authenticators have been received from said client application.

7. (previously presented) A method according to claim 23, comprising timing out said address token of an application of a currently authenticated user if no document request is received from said client application for a predetermined period.

8. (previously presented) A method according to claim 23, comprising authenticating said user for access to a plurality of Web servers located in the same Internet domain; and

enabling each of said Web servers to validate document requests from the client application, which requests include said address token, by checking said status data on receipt of a document request.

9.-22. (canceled)

23. (currently amended) A method of operating an authenticating server system for authenticating a user of a client application provided on a client terminal having no unique IP address via a data communications network, the server system being arranged to control access to a document stored on a resource server connected to said data communications network, said method comprising performing the following steps in said server system:

receiving at the resource server a request for said document generated by said client application provided on the terminal having no unique IP address;

evaluating at the resource server client-side persistent information accompanying said request including:

checking if the client-side persistent information contains an address token previously issued by the resource server which uniquely identifies the user, and performing the following steps at the resource server:

i) if no address token which uniquely identifies the user is contained in the client-side persistent information accompanying said request:

generating an address token which uniquely identifies the terminal address of the user, the generated address token replacing an IP address of the client terminal as a way of subsequently re-identifying the terminal address of the user;

transmitting the generated address token to the client application in a client-side persistent information packet so that the~~an~~ address token ~~which~~ can be used to uniquely re-identify~~ies~~ the user when ~~is generated and~~ re-transmitted with user authentication data to ~~without prior receipt at~~ the resource server ~~of a previously issued address token which uniquely identifies the user~~; and

storing said address token for the user; and~~or~~

ii) if an address token is received which accompanies user authentication data,

using said address token to uniquely re-identify~~ies~~ the address of the terminal from which the original document request was received ;~~the user is contained in the client- side persistent information accompanying said request and the address token is an unvalidated address token:~~

validating the address token using ~~other~~ said user authentication data received from the client terminal in said client-side persistent information ~~and~~ by reference to user authentication data already stored on said resource server;

updating ~~storing~~ the validated address token for an authenticated user with ~~and~~ an access status of the authenticated user associated with the validated address token;

transmitting a client-side persistent information packet containing the updated validated address token to the client terminal; and~~or~~

iii) if an address token which uniquely identifies the user is contained in the client-side persistent information accompanying said request and the address token is a

validated address token, using said validated address token to enable said resource server to validate said request for said document by checking if said stored access status for said user includes access to said document.

24. (previously presented) A method as claimed in claim 23, wherein step (ii) further comprises:

transmitting said requested document to said client terminal along with the client-side persistent information packet containing the validated address token to the client terminal.

25. (currently amended) A method of operating an authenticating server system connected to a data communications network, the server system being arranged to authenticate a user of a client terminal connected via the data communications network to a resource server to control access by the user to a document stored on said resource server, the client terminal having no unique IP address, said method comprising:

storing within the authenticating server system authentication details of authorized users of said resource server;

sending the user an unvalidated tag to enable subsequent re-identification of the terminal address of the user;

authenticating the user by performing:

receiving user authentication data with ~~and an~~ a returned unvalidated ~~identifying~~ tag at the resource server for the user from the client terminal having no unique IP address,

validating said authentication data by determining if said authentication data corresponds to equivalent stored authentication details, and if so:

updating the tag to a validated user identifying tag;

transmitting ~~issuing a~~the validated user identifying ~~identifying~~ tag ~~for the user~~ to said client terminal for storage thereon~~;~~, ~~transmitting the validated identifying tag to the client terminal,~~ the validated user identifying tag being arranged to enable the client terminal to retransmit the validated user identifying ~~identifying~~ tag with document requests directed at said resource server; and

storing at the resource server status data indicating said validated user identifying tag as identifying a terminal address and ~~of~~ a currently authenticated user; and

checking at the resource server said status data on receipt of a request for a document received from the client terminal, wherein if said resource server determines the request contains a validated user identifying tag, the authenticated user is given access to the system.

26.-28. (canceled)

29. (currently amended) The method as in claim ~~28~~25, wherein ~~said~~ if said authentication is invalid, a new identifying tag is issued, the new identifying tag comprising ~~comprises~~ data indicating the number of times an invalid authenticator and/or invalid authentication data have been received from said client terminal.

30. (currently amended) The method as in claim 2~~5~~9, wherein said method further comprises not issuing ~~no~~ a validated ~~further identifying~~ tag to said client terminal if ~~an identifying~~ tag received from said client terminal indicates that a predetermined number of invalid ~~identifying~~ tags and/or invalid authentication data have been received from said client terminal.

31. (previously presented)The method as in claim 25, comprising timing out said validated identifying tag as an identifying tag of a client terminal of a currently authenticated user if no document request is received from said client terminal for a predetermined period.

32. (previously presented) The method as in claim 25, comprising authenticating said user for access to a plurality of Web servers located in a same Internet domain; and

enabling each of said Web servers to validate document requests from the client terminal, which requests include said validated identifying tag, by checking said status data on receipt of a document request.

33. (previously presented) The method as in claim 25, wherein in said server system a plurality of documents are stored on a plurality of resource servers, wherein the step of validating the authentication data of a user comprises remotely authenticating the user by reference to authentication details of an authorized user stored by one of said plurality of resource servers, the remote authentication comprising:

generating status data to distinguish said user from other users who are not currently authenticated; and

generating a secret encryption key shared with said user, and wherein said method of operating the authentication server further comprises:

storing said status data in a storage device accessible to each of said plurality of resource servers to check an authentication status of said user by using said validated identifying tag for said client terminal received in said request; and

storing said shared secret key in a data store accessible by at least one of said resource servers for use during communications with said user.

34. (previously presented) The method as in claim 33, wherein said authenticating step comprises issuing a challenge to the client terminal, receiving a response to said challenge, and verifying said response.

35. (previously presented) The method as in claim 33, further comprising updating said status data for an authenticated user following said storing step in said storage device.

36. (previously presented) The method as in claim 35, wherein said updating step is performed in response to a time-out associated with said status data.

37. (previously presented) The method as in claim 35, wherein said updating step is performed in response to access by one of said resource servers to said status data.

38. (previously presented) The method as in claim 36, wherein said updating step is performed in response to a request by the client terminal.

39. (canceled)

40. (previously presented) The method as in claim 33, wherein said status data is stored in a data store which each of said resource servers are able to access.

41. (previously presented) The method as in claim 33, wherein said authentication details include data identifying the rights of access of individual users to one or more of said resource servers.

42. (currently amended) An authenticating server system adapted to perform the method of claim 25, wherein the client terminal supports a client application which the user uses to seek access to said document, and wherein the validated user identifying tag comprises a unique user identifying tag for the client application of the client terminal.

43. (previously presented) An authenticating server system adapted to perform the method of claim 25.

44. (new) A method of re-identifying a user when performing steps in an authentication method, the authentication method enabling the user to access a document requested via a terminal connected by an Internet Protocol connection to an application

server, wherein the user terminal individually has no unique IP address, said re-identifying method comprising:

receiving a document request from an application client of the user for access to said application server;

checking if the request contains a validated or unvalidated token enabling the terminal of the user to be subsequently re-identified, and if not,

providing an address token to the user in an initial cookie containing in a field an identifying tag which replaces the non-unique IP address of the terminal as a means for subsequently re-identifying the terminal,

storing at the application server said address token as an unvalidated replacement address token;

receiving authentication details from the user including a username and password with a returned unvalidated address token, whereby the terminal address is re-identified from the replacement address token.

45. (new) A method as claimed in claim 44, wherein the method further comprises performing the following steps to authenticate the user at the terminal by comparing stored authentication details with said received authentication details:

hashing said received user name and password at a secure password client;

passing on the hashed user name and password to a cache management server which polls a secure password server to check whether the username and password hash matches a stored value for an authorised user; and

if the user is authenticated,

the cache management server performing the steps of:

retrieving an access rights token from the secure password server, and if the access rights token indicates the user is allowed to access the application server to which the document request is directed,

sending an access allowed message to the application server, and

storing status data comprising: the access rights token, the validated address token and an authenticator comprising the hashed user name and password for the user, to collectively identify the terminal as a terminal of a currently authenticated user; and

at the application server performing the steps of,

receiving said access allowed message;

sending the requested document with an updated cookie containing:

the validated address token, and

the authenticator comprising the username and password hash; and

at the client terminal, storing the updated cookie comprising said validated address token, whereby said user is re-identified and authenticated when generating subsequent document requests by allowing said application server to access said status data at said cache management server in order to check the authentication status of a user on receipt of a document request containing said validated address token.

46. (new) A method as in claim 44, wherein the field is a name=value field.